SOFTWARE MAINTENANCE: RESEARCH AND PRACTICE
J. Softw. Maint: Res. Pract. **10**, 381–393 (1998)

Practice

Training for Software Maintenance

M. J. TAYLOR*, E. P. MOYNIHAN and A. LAWS
School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, L3 3AF, U.K.

SUMMARY

There are many factors which significantly affect the software maintenance work carried out in organizations. This paper concentrates on one of those factors which is the training that IT staff receive in software maintenance techniques. This paper examines the approaches to training for software maintenance found in case studies in 31 UK organizations. In particular, this paper identifies the formal and informal approaches to software maintenance training that were used within the organizations studied, examines how such training was carried out and discusses the perceived benefits of the different training approaches. © **1998 John Wiley & Sons, Ltd.**

KEY WORDS: software maintenance; training; IT training; business training; training approaches

1. INTRODUCTION

1.1. The nature of software maintenance

Software maintenance can be described as any deliberate change to an existing software system, e.g., to repair faults or keep it up-to-date. The various types of software maintenance are considered in publications such as Pressman (1992). An example of software maintenance is a change in payroll requirements. A UK illustration of this is the introduction in 1988 of non-taxable charitable contributions under Section 202 of the Income and Corporation Taxes Act 1988. In brief the legislation required employers, under certain conditions, to provide a facility for employees to have deductions made from pay as a contribution to registered charities, which would be free of tax up to a specified limit. This legislation required, for instance, changes in payroll software, business stationery and office procedures. See Tolley (1995) for details. Frequently it is not initially clear how such changes affect the overall business system. Indeed there may be a choice. For example, if some requirements are simple and infrequent then it may be more economic to handle them by amended manual office procedures, instead of changing the payroll software. This sort of maintenance decision presupposes some understanding of the existing business system in its entirety, as well as the new requirements.

* Correspondence to: M. J. Taylor, School of Computing and Mathematical Sciences, Liverpool John Moores University, Liverpool, L3 3AF, U.K. Email: m.j.taylor@livjm.ac.uk

CCC 1040–550X/98/060381–13$17.50

Received 23 June 1997
Revised 4 May 1998

It is often stated that software maintenance is up to 80 per cent of business IT work (Swanson and Beath, 1989). Thus, software maintenance work is of major economic importance, but this is not reflected in either publications or established information systems methodologies. In most cases software maintenance receives a brief mention and little more (Avison and Fitzgerald, 1995; SSADM, 1995). The emphasis in most textbooks and information systems methodology handbooks is usually on developing new systems.

In this paper we examine the training approaches that were used by 31 UK organizations to prepare their IT staff for software maintenance work. Any organization that intends to employ IT staff for software maintenance work can either arrange to train such staff themselves (by in-house or external commercial training) or recruit IT staff that have received training elsewhere (either in education or in previous employment). However, only one organization encountered in this research project, a financial services company, had a stated IT recruitment policy that only IT staff with previous knowledge of the particular application areas related to the vacancy would be employed.

This paper is aimed at both academics and IT practitioners. This paper intends to provide an empirical academic investigation of the training approaches for software maintenance that are used in current practice in the UK and their perceived benefits. In addition, for IT practitioners, this paper intends to provide a discussion of the training approaches for software maintenance that can be used by an organization, together with an indication of the potential benefits to the organization of such approaches.

Blumenstalk Mingus (1997) has commented that software legacy systems have unique support and maintenance requirements that can only be met by competent and specialized personnel who understand the systems' value and participate in individualized cross-training programmes. Pigoski and Looney (1993) stated the need for system-specific software maintenance training. Butler (1994), however, noted that most IT managers budget little, if any, money for software maintenance training and maintenance tools. Calliss and Calliss (1994) commented that there are few training programs or academic courses aimed at developing the skills that are required for software maintenance. However, Taylor, Moynihan and Wood-Harper (1997) and Zeltmann and Vineyard (1992) have noted the large skill set that is actually required for software maintenance work.

There are, however, government and professional bodies that provide guidelines for IT training, and these include software maintenance-specific advice. For example, the BCS industry structure model (release 3) (BCS, 1996) advises upon the training requirements for all IT activities, and in particular, specifies the training required for application support within the context of systems development and maintenance. The TickIT scheme also gives advice on IT training for both software development and maintenance (Wilson, 1994). Any organization that intends to apply for BS5750/ISO 9000 should certainly have records of the training requirements of their staff based on education, skills, qualifications and experience (Daily, 1992).

The originality of this paper is that there is currently no academic or commercial literature available that provides an empirical examination of the training approaches that are used in current practice for software maintenance, and their perceived benefits. Given the economic importance of software maintenance (for example, the costs associated with the Year 2000 problem (BCS, 1997)) any means for potentially improving software maintenance activities is worthy of investigation.

Table 1. Summary of organizations studied and selected responses

Type of organization	No. of employees (approx.)	No. of IT staff (approx.)	IT infrastructure	Business term. problem	Impact analysis problem	Second policy	Bus. tours	Bus. doc.	CASE tools	Bus. qual.	Software maintenance techniques
Engineering	8 000	50	Mframe, mini,PC	Y	N	N	Y	N	N	N	N
Computer services	90	35	Mframe, mini,PC	N	N	N	N	N	N	N	N
Engineering	350	7	mini	N	N	N	Y	N	N	N	N
Manufacturing	1 600	18	Mframe,PC	N	N	N	N	N	N	N	N
Financial services	4 000	160	Mframe,PC	Y	Y	Y	Y	Y	Y	Y	Y
Financial services (subsid.)	240	12	Mframe,PC	Y	Y	N	Y	Y	Y	N	Y
Financial services (subsid.)	120	10	Mframe,PC	Y	Y	N	Y	Y	Y	N	Y
Financial services (subsid.)	50	8	Mframe,PC	Y	Y	N	Y	Y	Y	N	Y
Financial services (subsid.)	40	7	Mframe,PC	Y	Y	N	Y	Y	Y	N	Y
Financial services	3 000	110	Mframe, mini,PC	Y	Y	N	Y	Y	N	Y	Y
Public utility	7 500	120	Mframe, mini,PC	N	Y	N	N	Y	N	N	N
Manufacturing	2 500	40	Mframe,PC	Y	N	N	N	N	N	N	N
Computer services	300	220	Mframe, mini,PC	N	N	N	N	N	N	N	N
Manufacturing	430	6	mini,PC	N	N	N	N	N	N	N	Y
Financial services	3 400	80	Mframe, mini,PC	Y	Y	N	N	Y	Y	Y	Y
Manufacturing	90	2	PC	N	N	N	N	N	N	N	N
Distribution	15 000	30	mini,PC	N	N	Y	N	N	N	N	N
Defence	7 500	40	Mframe, mini,PC	Y	N	N	N	N	N	N	N
Local government	30 000	40	Mframe, mini,PC	Y	N	N	N	N	N	N	N

Continued

2. RESEARCH METHOD

The research reported in this paper concerns the approaches to training for software maintenance found in case studies in 31 UK IT departments. The types of organizations researched are detailed in Table 1. The case studies were conducted by interviewing at least two members of IT staff in each of the organizations studied, in addition to inspecting

Table 1. Continued.

Type of organization	No. of employees (approx.)	No. of IT staff (approx.)	IT infrastructure	Business term. problem	Impact analysis problem	Second policy	Bus. tours	Bus. doc.	CASE tools	Bus. qual.	Software maintenance techniques
Higher education	2 000	18	mini,PC	N	N	N	N	N	N	N	N
Distribution	9 000	40	mini,PC	N	N	N	N	N	N	N	Y
Manufacturing	2 100	18	mini,PC	N	N	N	N	N	N	N	N
Civil service	3 000	320	Mframe, mini,PC	N	Y	N	N	N	N	N	Y
Local Government	80	6	mini	N	N	N	N	N	N	N	N
Retail	4 500	140	Mframe, mini,PC	Y	Y	N	N	N	N	N	Y
Distribution	35	2	PC	N	N	N	N	N	N	N	N
Manufacturing	1 200	40	Mframe,PC	Y	N	N	N	N	N	N	N
Distribution	1 500	30	mini,PC	N	N	N	N	N	N	N	N
Financial services	3 800	90	Mframe,PC	Y	N	N	N	Y	N	N	Y
Manufacturing	2 300	15	mini,PC	N	N	N	N	N	N	N	N
Manufacturing	1 400	18	mini	N	N	N	N	N	N	N	N
Percentage of organizations answering 'Yes'				42%	32%	7%	26%	29%	19%	10%	39%

Business term. problem — the organization cited problems with understanding business terminology.
Impact analysis problem — the organization cited problems with impact analysis in software maintenance projects.
Second. policy — the organization had a formalized secondment policy for IT staff.
Bus. tours — business tours were provided for new IT staff.
Bus. doc. — Business-orientated systems documentation was used.
CASE tools — CASE tools were used for software maintenance projects.
Bus. qual. — the organization encouraged IT staff to obtain relevant business qualifications.
Software maintenance techniques — the organization's IT methodology or IT standards contained specific software maintenance techniques.

available relevant documentation. The job titles of those interviewed varied between the organizations researched, but included, for example: analyst programmer, systems analyst, project manager and IT manager. The detailed research approach was as follows: contact was made with IT staff in the 31 organizations studied by one of two routes. The first route was by direct phone call to the organization's IT department, the other was through research seminars organized by the researchers. The research seminars were advertised by mail to the IT departments of local organizations. In either case the researchers arranged (through the established contact person within each organization) to visit the organization's IT department. Whilst within the IT department the researchers conducted semi-structured interviews with at least two of the IT staff, and viewed relevant organizational documentation where available. By this mechanism the researchers sought to answer the following research questions:

(1) What aspects of software maintenance require formal training?

(2) What aspects of software maintenance require informal training?
(3) What training does your organization provide for software maintenance?
(4) Who provides training for software maintenance in your organization?
(5) When is the training for software maintenance provided for IT staff during their career?
(6) How is the training for software maintenance carried out in your organization?
(7) What are the benefits of training for software maintenance?

With these questions the researchers intended to examine the actual approaches that were used for software maintenance training within the organizations studied; to examine how such training was carried out; to identify the formal and informal aspects of such training, and to attempt to assess the perceived benefits of the training approaches used.

The objectives of the investigation were to provide an empirical academic study of the approaches used for software maintenance training in current practice within the UK, and their perceived benefits, and in addition to provide IT practitioners with a discussion of available software maintenance training approaches and their potential benefits for an organization.

The case study approach is appropriate for this research project since it allows the researchers to gain a detailed understanding of the training approaches that can be used for training in software maintenance, and enables the researchers to gain a more holistic perspective of the research area as described by Gummesson (1991). In particular the case study approach can overcome some of the problems of terminology differences between organizations that can bedevil other more superficial research approaches such as questionnaires. Benbasat, Goldstein and Mead (1987) point out that case studies are particularly beneficial in research areas (such as the training for software maintenance) where there is no established theoretical base.

3. RESEARCH RESULTS

During the course of this research, the researchers identified a number of approaches that were used for providing training for software maintenance by the 31 organizations researched. Most of these approaches were in limited use, and no organization used them all. If they were being used, these approaches were considered reasonably effective by the IT practitioners interviewed. If not used, they were usually considered a useful idea for future training.

3.1. Formal training approaches for software maintenance

The formal training approaches for software maintenance cited by the IT practitioners interviewed in the 31 organizations studied included:

(1) Training courses

In the course of conducting the case studies in the 31 organizations researched, the researchers found no examples of training courses for IT staff aimed specifically at

software maintenance. The only training courses provided for IT staff in the 31 organizations researched that included any aspect of software maintenance were a small number of external commercially-provided courses that included regression testing as part of software testing in general. Noordam (1996) has noted that the lack of training courses in software maintenance was a significant problem in the Netherlands.

(2) Business qualifications

This involved IT staff obtaining business qualifications in areas relevant to their employer. This approach to training for the business aspects of software maintenance work was encouraged in only 3 of the 31 organizations studied, and all were in the financial services sector. The few IT staff involved were encouraged to obtain financially-orientated qualifications offered by professional bodies, for which the organization met the costs involved. Chapin (1987) has suggested that maintainers should learn some of the theory and practice associated with the area of the organization that the software serves, but has not investigated mechanisms for achieving this. Less demanding approaches than obtaining professional qualifications can be used, for example, attending one-day business courses a couple of times per year.

(3) Secondment to user departments

A systems analyst or analyst programmer can be seconded to a business department, perhaps for a few days. During this experience the analyst or analyst programmer can absorb many practical details by actually performing the same tasks as the application software users. This can provide a background in business operations like stock control. It should improve the understanding of software maintenance requests from application software users (Taylor, Moynihan and Wood-Harper, 1997; Taylor, 1997). This approach can also uncover both underlying problems and opportunities to enhance systems (Tayntor, 1989). In some cases systems analysts can do normal IT work mostly within the business department rather than the IT department. Only 2 of the 31 organizations researched had any formalized secondment policy for IT staff, as shown in Table 1. The cost to the IT department of such training was the portion of the salaries of the IT staff consumed whilst they were in the user department, and therefore unavailable for IT work. In one medium-sized distribution company researched, roughly a quarter of the systems analysts employed by the company were permanently resident in the business user departments.

3.2. Informal training approaches for software maintenance

Informal training approaches for software maintenance were divided between application-specific training which included: reviewing maintenance documentation, transferring knowledge from experienced staff, reviewing business documentation, and expert systems and general maintenance training which included: CASE tools and maintenance-specific aspects of the IT methodology/IT standards in use.

(1) Reviewing maintenance documentation

The IT staff in all the 31 organizations studied indicated that relevant systems documentation would be reviewed as part of a typical software maintenance project. The state of systems documentation was difficult to quantify, and, in the main, qualitative descriptions of the systems documentation within the 31 organizations studied was all that could be obtained. The quantity and quality of documentation varied by organization and within an organization by system, and even by software module. In some of the organizations studied virtually no systems documentation was available, and IT staff worked almost entirely from the source code. In the majority of the 31 cases researched, systems documentation was declared to be out-of-date. As an example, in a large financial services organization studied, a database administrator confirmed that the data dictionary in use in the IT department was less than half populated, as IT staff did not bother to update it properly during maintenance projects.

Examination suggested that most systems documentation was more development-orientated than maintenance-orientated. The production of maintenance-orientated documentation was occasionally found, but incomplete. Maintenance-orientated documentation can, for instance, specify known 'hot-spots' which are frequently amended. Another example of maintenance-orientated documentation was the provision of written guidance on recurring similar software maintenance tasks.

Documentation can cover the business aspects of an application system as well as the physical systems design. However, during this research, business-orientated documentation was seen in only nine of the organizations studied. Table 1 shows those organizations that used business-orientated documentation. Most documentation, when it existed, concerned the physical system.

The cost of reviewing maintenance documentation in order to train IT staff in the workings of a given application system was absorbed within the software maintenance projects undertaken upon the given application system.

(2) Transferring knowledge from experienced staff

This research found that IT maintenance staff often needed to consult 'experts' within the IT or business departments, usually on an *ad hoc* basis. More structured approaches to training IT maintenance staff by transferring knowledge from experienced staff included:

(i) Presentations, e.g., a software designer explaining the physical design of a large IT insurance system. Surprisingly this was only used for proposing future projects and not explaining existing systems.
(ii) Discussions, e.g., a warehouse manager discussing how a computer stock control system is used within his or her department. This approach was rarely used in any of the 31 organizations studied.
(iii) Tours, e.g., with a large order processing system, programmers can be given a tour of the sales office and warehouse to obtain an overview of the whole business system. Business tours were provided for new IT staff in 8 of the 31 organizations studied. No recent examples of tours being given to established IT staff, were found.

Presentations and discussions can also be used for specific maintenance projects, although this approach was rarely used by any of the 31 organizations studied. Younger and Bennett (1993) have noted, however, that the understanding of a given application system gained by IT maintenance staff was rarely formally recorded for the benefit of their successors.

(3) Reviewing business documentation

The collection of internal business documentation, e.g., company brochures or procedures manuals, is a standard systems analysis technique. External documentation, e.g., legal regulations, can also be important. An example for payroll IT systems is CA29 (the manual for employers on Statutory Maternity Pay published by the DSS Contributions Agency). All the 31 organizations studied seemed to be poor at utilizing business documentation for software maintenance projects on a regular basis. Systems analysts in effect seemed to rely on business user staff to identify and translate business literature into software maintenance requirements. The IT practitioners interviewed within all the 31 organizations studied indicated that they would typically only consider using business documentation for a software maintenance project if they could not obtain the relevant information through the business users.

(4) Expert systems for software maintenance training

For very large and complex application software systems with a long life expectancy it could, in theory, be economically viable to develop expert systems to help train new IT staff in the maintenance of the given systems (Cousin and Collofello, 1992). Layzell and Macaulay (1994) concluded that such expert systems would be particularly useful in organizations with rapid staff turnover or where there is an inability to attract IT staff. Such expert systems could contain the rules and heuristics, that are in present practice, contained in the minds of business and software system experts, as well as what is contained in documentation. The MACS (maintenance assistance capability for software) tool set (Georges, 1992) is one example of such an expert system. However, none of the 31 organizations studied had such expert systems, and only one of the organizations, a large financial services company, expressed any interest in such expert systems.

(5) CASE tools

CASE tools specifically aimed at the software maintenance environment can, in theory, assist in the training of IT maintenance staff. For example, CASE tools can help to train IT staff in the maintenance of a given application system, e.g., by making the systems structure, control flow and data usage more visible to the maintainer. However, reservations about CASE tools were expressed in all the organizations researched. Only six of the organizations studied used any form of CASE tool for software maintenance, as shown in Table 1, and then only in a limited way. Typically this was limited to data dictionary and debugging assistance for software maintenance staff.

(6) Maintenance-specific aspects of the IT methodology/standards

During the course of this research the researchers found that few of the information systems methodologies or IT standards in use in the 31 organizations studied included much in the way of maintenance-specific techniques. Only 12 of the 31 organizations studied had any techniques specifically aimed at software maintenance. For example, in one manufacturing company researched, the maintenance-specific techniques consisted purely of a maintenance request specification, a test plan and test case procedure, and an authorization to go live procedure. A result of the lack of software maintenance-specific techniques was that there was often no defined means of training IT staff in those activities that should typically be undertaken in a software maintenance project. A more detailed discussion of the information systems methodologies in use in the organizations studied and the way in which they were used for software maintenance activities can be found in Taylor and Wood-Harper (1996).

3.3. Software maintenance training during IT career

One of the avenues of investigation followed by the researchers was to ascertain when within their career paths IT staff received training in software maintenance. Formal training approaches, such as training courses and business qualifications, were aimed more at new starters within the 31 IT departments studied. In general, secondment to user departments usually occurred within the first few years of joining an IT department, in the case of the two organizations that had a formalized secondment policy. However, informal software maintenance training tended to take place throughout the career of those IT staff interviewed in the 31 organizations studied. Informal software maintenance training typically took place in the course of carrying out actual software maintenance projects in the 31 organizations studied, and was universally aimed at non-managerial IT staff.

3.4. Perceived benefits of software maintenance training

Nearly all the IT practitioners interviewed in the 31 organizations researched agreed that software maintenance training would be beneficial both to themselves and the organizations employing them. Specific benefits of software maintenance training quoted by those interviewed included:

(1) Reduced software maintenance project lead times

The vast majority of the IT practitioners interviewed indicated that the more they knew about an application area, and the workings of a given application system within that area, the easier it was to conduct maintenance projects on the given application system, which should reduce maintenance project lead times. In particular, software maintenance training in the actual application system (provided by either reviewing maintenance documentation, consulting IT application ‘experts’, or via CASE tools) was perceived as assisting in reducing the effort required for impact analysis. Abbattista *et al.* (1994) have noted that appropriate maintenance documentation can help to improve impact analysis.

Business-orientated training (provided by either business qualifications, secondment to user departments, business tours or reviewing business-orientated documentation) was perceived as assisting in the reduction of the problems associated with understanding business terminology. Layzell and Macaulay (1994) have suggested that business (or domain) knowledge can be essential to the success of the software maintenance function. The organizations quoting problems with impact analysis and understanding business terminology are shown in Table 1.

(2) More thorough testing

The main problem indicated by the IT staff interviewed with testing in software maintenance projects was impact analysis, that is, attempting to determine which parts of a given application system will be affected by the project. The organizations that cited impact analysis as a problem area are shown in Table 1. Application-specific training via maintenance documentation, IT application 'experts' or CASE tools was viewed by the IT practitioners interviewed in those organizations as a means of promoting more thorough testing.

(3) Better communication with users

The problem of understanding business terminology was encountered in 13 of the 31 organizations researched. Training in the business aspects of specific application areas (provided by either business qualifications, secondment to user departments, business tours or business-orientated documentation) was seen as one approach for reducing this problem by the IT practitioners interviewed.

It should be noted that it is not necessarily just IT staff who have a need for training in business operation (Arfa, Mili and Sekhri, 1991). This research indicated that for companies having a high turnover of business user staff, the business knowledge of the user community may significantly diminish. For example, in the eight financial services organizations researched, all of which sold insurance policies, the IT practitioners interviewed in those organizations indicated that few of the business user staff were aware of how insurance policy premiums were calculated, as the application software systems performed this activity for them. In a case study reported by Abran and Nguyenkim (1993) in a Canadian financial institution, the users had requested details of computerized algorithms based on either the systems documentation or on the program code. They could not obtain this through their own records. In other words, the users did not know the details of their own 'business logic'.

None of the 31 organizations studied actually analysed the impact of any software maintenance training that they provided for their IT staff. Mead, Tobin and Couturiax (1996) have noted that very few organizations analyse the effect of training provided for their IT staff.

CONCLUSIONS

In this paper we have examined the various approaches that were used for training for software maintenance based on case studies in 31 UK organizations. Our general conclusions, based on the research results are as follows.

Software maintenance training appeared to be a neglected area of the IT provision within the 31 organizations studied. Little effort appeared to be put into providing specific training for IT maintenance staff in the skills they required.

Business qualifications (used in only 10 per cent of the organizations studied), secondment to user departments (used in only 7 per cent), business tours (used in 26 per cent) and business-orientated documentation (used in 29 per cent) were perceived to improve the IT practitioner's business knowledge (Taylor, Moynihan and Wood-Harper, 1997) relating to the business areas they supported. The perceived benefit of this was an improvement in communications with business users, which appeared to help to reduce software maintenance project lead times.

Reviewing maintenance documentation, consulting IT application 'experts' (both used in all the organizations studied) and CASE tools (used in only 19 per cent) were perceived to assist in improving testing (particularly with regard to impact analysis, which was quoted as a problem in 32 per cent of the organizations studied). This also was perceived as helping to reduce software maintenance project lead times.

This research indicated that there was no overall 'best' approach for providing software maintenance training, but rather that different training approaches could be used to enhance the skill set required by IT maintenance staff (Taylor, Moynihan and Wood-Harper, 1997). However, it was perceived by the IT practitioners interviewed that specific training approaches could be used to improve problematic areas of software maintenance projects, in particular understanding business terminology and impact analysis.

Further empirical research is required in this area in order to develop software maintenance training strategies, which can potentially benefit organizations by helping to reduce the high costs of software maintenance activities.

References

Abbattista, F., Lanubile, F., Mastelloni, G. and Visaggio, G. (1994) 'An experiment on the effect of design recording on impact analysis', in *Proceedings of the International Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos CA, pp. 253–259.

Abran, A. and Nguyenkim, H. (1993) 'Measurement of the maintenance process from a demand-based perspective', *Software Maintenance: Research and Practice*, **5**, 63–90.

Arfa, L. B., Mili, A. and Sekhri, L. (1991) 'An empirical study of software maintenance', in *Proceedings of the Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos, CA, pp. 52–58.

Avison, D. E. and Fitzgerald, G. (1995) *Information Systems Development: Methodologies, Techniques and Tools*, McGraw Hill, Maidenhead.

BCS (1996) *British Computer Society Industry Structure Model (Version 3)*, The British Computer Society, Swindon.

BCS (1997) *The Year 2000: A Practical Guide for Professionals and Business Managers*, British Computer Society, Swindon.

Benbasat, I., Goldstein, D. and Mead, M. (1987) 'The case research strategy in studies of information systems', *MIS Quarterly*, September, 369–386.

Blumenstalk Mingus, N. (1997) 'The legacy staffing challenge', *Information Systems Management*, **14**(1), 22–25.

Butler, J. (1994) 'Software maintenance schizophrenia', *Managing Systems Development*, **14**(12), 1–3.

Calliss, F. and Calliss, D. (1994) 'Suggested scenarios of software maintenance education', in *Proceedings of the 7th SEI CSEE Conference*, Springer-Verlag, Berlin, pp. 329–340.

Chapin, N. (1987) 'The job of software maintenance', in *Proceedings of the Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos CA, pp. 4–12.

Cousin, L. and Collofello, J. S. (1992) 'A task based approach to improving the software maintenance process', in *Proceedings of the Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos CA, pp. 118–126.

Daily, K. (1992) *Quality Management for Software Maintenance*, NCC Blackwell, Oxford.

Georges, M. (1992) 'MACS: maintenance assistance capability for software, *Software Maintenance: Research and Practice*, **4**, 199–213.

Gummesson, E. (1991) *Qualitative Methods in Management Research*, Sage, New York.

Layzell, P. and Macaulay, L. (1994) 'An investigation into software maintenance perception and practices', *Software Maintenance: Research and Practice*, **6**, 105–120.

Mead, N., Tobin, L. and Couturiax, S. (1996) '*Best training practices within the software engineering industry*', Technical report CMU/SEI-96-TR-034, Software Engineering Institute, Carnegie Mellon University, Pittsburgh.

Noordam C. (1996) 'Why is maintenance a dirty word?', *Informatie*, **38**, 29–33.

Pigoski, T. and Looney C. (1993) 'Software maintenance training: transition experiences', in *Proceedings of the Conference on Software Maintenance*, IEEE Computer Society Press, Los Alamitos CA, pp. 314–318.

Pressman, R. (1992) *Software Engineering: A Practitioners Approach*, McGraw-Hill, Maidenhead.

SSADM (1995) *SSADM 4+ Reference Manual*, NCC Publications, Manchester.

Swanson, E. and Beath, C. (1989) *Maintaining Information Systems in Organisations*, John Wiley, Chichester.

Taylor, M. J. (1997) *Financial Times Management Briefings: Managing Systems Maintenance*, FT Pitman Publishing, London.

Taylor, M. J. and Wood-Harper, A. T. (1996) 'Methodologies and software maintenance', *Software Maintenance: Research and Practice*, **8**(5), 295–308.

Taylor, M. J., Moynihan, E. and Wood-Harper, A. T. (1997) 'Knowledge for software maintenance', *Journal of Information Technology*, **12**(2), 155–166.

Tayntor, C. B. (1989) 'When (and how) to replace a system', *Systems Development*, **9**(12), 5–10.

Tolley (1995) *Tolley's Payroll Handbook*, Tolley Publishing, Croydon.

Wilson, M. (1994) 'TickIT and consultants: how to get IT without them', in *Proceedings of the Conference on Software Quality Management*, Computing Mechanics Publications, Southampton, pp. 115–123.

Younger, E. J. and Bennett K. H. (1993) 'Model-based tools to record program understanding', in *Proceedings of the IEEE Workshop on Program Comprehension*, IEEE Computer Society Press, Los Alamitos CA, pp. 87–95.

Zeltmann, S. and Vineyard, M. (1992) 'A review of research in progress: an investigation into skills required to successfully maintain existing software', in *Proceedings of the ACM SIGCPR Conference*, ACM Press, New York NY, pp. 388–392.

Authors' biographies:

Mark Taylor is currently a Senior Lecturer at Liverpool John Moores University. In the course of his industrial career he has worked in both the manufacturing and finance sectors as an Analyst Programmer, Systems Designer and Systems Analyst. He is a member of the British Computer Society and an active IT consultant. He is the author of a book on systems maintenance.

Eddie Moynihan is currently a Reader in Information Technology for business administration at Liverpool John Moores University. He spent several years as an Analyst Programmer in manufacturing and commerce, and was a Project Manager for a systems house in charge of developing financial software packages. He was also a systems development manger with the Dunlop Group. He is the author of a book on business systems analysis and is a member of the British Computer Society and a Chartered Engineer.

Andy Laws is currently a Lecturer in Information Systems at Liverpool John Moores University. He was previously an IT training specialist in a large commercial organization. He is currently undertaking a Ph.D. in software maintenance.